UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/449,782	11/26/1999	JAMES MCKEETH	MICE-0089	6698

7590 01/24/2005

COE F MILES
TROP PRUNER HU & MILES P C
8554 KATY FREEWAY
SUITE 100
HOUSTON, TX 77024

EXAMINER	
STEELMAN, MARY J	
ART UNIT	PAPER NUMBER
2122	16

DATE MAILED: 01/24/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/449,782	Applicant(s) MCKEETH, JAMES	
	Examiner Mary J. Steelman	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 24 September 2004.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-22 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-22 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. Based on a decision by the Board of Appeals, the previous Final Office Action has been withdrawn. New art has been discovered and applied hereto, in combination with US Patent 6,182,279 to Buxton, previous art of record. Claims 1-22 are pending.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over U. S. Patent No. 6,182,279 to Buxton, in view of "Windows 95 SECRETS 3rd Edition" by Brian Livingston & Davis Straub (hereinafter Livingston).

Regarding claim 1, Buxton teaches:

-receiving an identifier. (Col. 12, lines 2-3: "User may enter a descriptive name (identifier) of the template..." Also, col. 9, lines 5-7, "Components register themselves...so that the OLE libraries are aware of the component's existence." As an example of "receiving an identifier", refer to col. 12, lines 18-21, "A menu...is presented upon selection of the Create Distribution Pack option and allows users to identify one or more templates to be packaged...")

-storing the command output in a system storage at a location identified by the identifier. (Col. 11, lines 17-18, "...these templates may be stored to facilitate efficient distribution of templates to others...", "Col. 13, lines 10 – 14: "Template storage DLL (system storage) performs a number of ...storage/retrieval methods..." (underlining added) Also see col. 14, lines 7-12, "When template builder saves the template, it instructs the component, to save itself as an OLE structure storage file. Additional data to identify and register the component is also saved...") Buxton disclosed a template builder utility (col. 2, lines 32-34) that allows user selected options to modify base components (OLE controls) and stores the modifications as templates.

Buxton suggested:

-A method to provide process command line utility output comprising:

Buxton, col. 8, lines 45-53, disclosed command line processes (command line utility output) as an alternative to a user interface used to modify a system. "A user interface enables a user to interact with component system and may be implemented with a simple command line interpreter (command line utility)...The design of such a user interface is within the scope of those reasonable skilled in the arts."

-receiving output from a command line utility;

Buxton suggested: (Col. 13, line 8:) "Template builder creates templates..." Also see col. 2, lines 30-38, which recites, "A template builder utility allows a user to select a base component, modify the persistent data...and store (the received output) the modifications as a template in a predetermined format. The template builder utility further enables the user to package templates (received output from command line) in a template distribution package format which enables the templates to be distributed…")

Buxton suggested receiving commands via command line, which results in modifying the registry (system storage) and storage. Buxton failed to specifically disclose that the command line input used a "command line utility". Buxton suggests that the command line input (an object that consists of modifications to base component) is directed (DIR utility) to storage, and the registry is edited (the REGEDIT utility), but did not explicitly disclose 'command line utility'.

However Livingston explicitly disclosed (Page 315, second half of page, see "The DOS Version of the Registry Editor") using a command line utility to edit the registry. "It is possible to edit the Registry (utility to edit system storage) **from the DOS command prompt** (command line) . (emphasis added) The **utility** "REGEDIT" takes arguments (supplied by switches – See code segment, bottom of page 315, "The DOS Regedit syntax is as follows: As an example: /L:system Specifies the location of the System.dat file.) that specify the location of the System.dat file (/L: system), the User.dat file (/R: user), the file to import into the Registry (filename1: receiving an identifier), etc. (emphasis added) Using this **command line utility**, output is stored in the registry (system storage) at a location identified by the identifier. (emphasis added). It should be noted that DOS command prompt is used to enter a command to invoke a utility. Command-line utilities are an alternative way to start code execution. The functionality is the same, whether you start from graphical user interface or from a command line utility.

The following dictionary definitions further support the rejection:

As defined in Microsoft Computer Dictionary, 5th Edition, page 111, **command line**: A string of text written in the command language and passed to the command interpreter for execution. (emphasis added) As defined in Microsoft Computer Dictionary, 5th Edition, page 544, **utility**: A program designed to perform a particular function; the term usually refers to software that solves narrowly focused problems or those related to computer system management. (emphasis added)

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Buxton's GUI to include the more primitive command line interpreter utility for effectively accessing and/or interacting with system storage as suggested by Buxton (col. 8, lines 45-47). Furthermore, by providing specific details, as disclosed by Livingston, regarding modifying system storage (the registry) by using such a DOS command prompt, such as REGEDIT(command line utility) , which when augmented with switches, redirects the received output to be stored at a location identified by the identifier, because these are defined language commands used to provide options to a Windows language programmer for customizing the registry as needed for initialization, enhancing accessibility when a graphical user interface is not available to support an executing program.

Regarding claim 2, Buxton teaches:

-receiving an identifier that identifies one or more entries in a system registry database. (Fig. 2, item 205 and col. 13, lines 14-15, "...registry keys are created..." Also see col. 14, lines 29-59, "To facilitate loading of template onto another system...a number of registration key or subkey are included with template. Each template may have the keys 450A-I, as illustrated in Fig. 4C...Key 450H contains information indicating the name of the storage object in template storage file where initialization data...may be located...Key 450I contains information identifying the CLSID...)

Regarding claim 3, Buxton teaches:

-receiving a root key identifier. (Col. 11, line 2: "Most OLE object application information is stored in subkeys under the CSLID root key..." Also see col. 17, lines 35-41, "Component loader loads, verifies and checks the license of a component by replacing in registry the InProcessServer 32 entry, i.e. key 450A...and adding additional registry keys 450B-J, as previously described, that will let the component loader (receiving a root key identifier) then load the correct OLE control.")

Regarding claim 4, Buxton teaches:

-receiving a sub-key identifier. (Col. 11, line 2 and col. 14, line 31: To facilitate loading of template...a number of registration or subkey are included with template...")

Regarding claim 5, Buxton teaches:

-a WINDOWS operating system registry database. (Col. 4, line 49: "Operation of computer system is generally controlled...by operating system software, such as...Windows95...")

Regarding claim 6, Buxton teaches:

-receiving a system storage identifier. (Col. 12, lines 20-21, "...users identify...templates to be packaged..." Also for another example of receiving a system storage identifier, see col. 20, lines 42-45, "...relevant character string from the registry is converted to CLSID. The component loader (receives a system storage identifier) then calls the GetClassObject to retrieve the real component's class factory...")

Regarding claim 7, Buxton teaches:

-receiving an identifier indicating a system registry. (Col. 10, line 66 – col. 11, line 4: A CLSID identifies the functionality of an object class that can display...access to property values...A subkey is used by an OLE to find out information about the control.")

Regarding claim 8, Buxton teaches:

-receiving an identifier indicating shared system memory. (Col. 8, lines 6-7: "OLE libraries (shared) comprise the set of system-level services in accordance with the OLE specification...")

Regarding claim 9, Buxton teaches:

-shared system memory identifies a system clipboard memory. (Col. 11, line 6: "An FORMATETC...is an OLE data structure which acts in a generalized clipboard format...")

Regarding claims 10 and 11, Buxton teaches:

-receiving output directly from the command line output utility.

-receiving output from the command line output utility through a subsequent command line output routine.

(Col. 2, lines 30-34, "A template builder utility allows a user to select a base component, modify the persistent data of the component, and store the modifications as a template in a predetermined format." (storage receives modified templates) Also, col. 2, lines 34 – 38 and col. 13, lines 23-24: "The template builder utility further (subsequent command line output routine) enables the user to package (by receiving) templates (output for command line utility) in a template distribution package format which enables the templates to be distributed to other users..." Also see, col. 14, line 61-65, "...the Create Distribution Pack option...")

Regarding claim 12, Buxton teaches:

-associating each line of command line utility output with a line identifier in the system storage. (Col. 3, lines 1-9: Template storage with a means for indexing, including key information associated with the template. "...a memory having one or more locations, means for indexing one or more locations within the memory..." Also col. 13, lines 35-44, templates are stored with an enumerated decimal number: "Each template is stored in an ISTORAGE whose name is unique...and may have the form TEMPLEnnn, where nnn may be a decimal number.")

Regarding claim 13, Buxton teaches:

-setting each line identifier to a value corresponding to that line's position in the command line utility output. (Rejection of claim 12 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 13 is rejected under the same rational as claim 12.)

Regarding claim 14, Buxton teaches:

-setting a default value of the received identifier to equal the total number of command utility output lines stored in the system storage. (Rejection of claim 12 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 14 is rejected under the same rational as claim 12.)

Regarding claim 15, Buxton teaches:

Note: Buxton, col. 8, lines 51-53, disclosed command line processes as an alternative to a user interface.

-A program storage device (Col. 2, line 49- 52.)

Claim 15 contains limitations as recited in claim 1, therefore claim 15 is rejected under the same rational as claim 1.

Buxton disclosed receiving commands via command line which result in modifying the registry. Buxton failed to specifically disclose that the command line input used a "command line utility". It should be inherent that the command line input (an object that consists of modifications to base component) is directed (DIR utility) to storage, and the registry is edited (the REGEDIT utility).

However Livingston and Straub explicitly disclosed (Page 315, second half of page) using a command line utility to edit the registry. The utility "REGEDIT" takes arguments (supplied by switches) that specify the location of the System.dat file, the User.dat file, the file to import into the Registry (filename1: receiving an identifier), etc. Using this command line utility, output is stored in the system storage (registry) at a location identified by the identifier.

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to have modified Buxton's GUI to include the more primitive command line interpreter utility for input to system storage as suggested by Buxton (col. 8, lines 45-47). Furthermore, by providing specific details, as disclosed by Livingston & Straub, regarding modifying system storage (the registry) by using a DOS command prompt (command line utility), REGEDIT, which when augmented with switches, redirects the received output to be stored at a location identified by the identifier, because these are defined language commands used to provide options to a Windows language programmer for customizing the registry as needed for initialization.

Regarding claim 16, Buxton teaches:

-instructions to store command line utility output in an operating system registry database. (Col. 13, lines 10 – 15: "Template storage DLL (operating system registry database) performs a number of formatting and storage/retrieval methods...Template storage DLL ensures all additional registry keys...are created...")

Regarding claim 17, Buxton teaches:

-instructions to store command line utility output in an operating system maintained volatile memory. (Fig. 1, item 110 (volatile) and col. 23, line 36 – col. 24, line 10: "A software implementation of the above described embodiment may comprise a series of computer instructions either fixed on a tangible medium...or transmittable to a computer system...such instructions may be stored using any memory technology ...on system ROM or fixed disk...")

Regarding claim 18, Buxton teaches:

-instructions to receive one or more lines of output from the command line utility. (Fig. 2.)

-instructions to store each of said one or more lines of output in the system storage. (Fig. 2, item 205, col. 14, lines 26-29: "The remainder of the operating system registry entries are generated by code (instructions to store) in the template storage DLL and are stored in both registry and the template.")

Regarding claim 19, Buxton teaches:

-instructions to associate a unique identifier with each of the one or more lines of output stored in the system storage. (Col. 13, lines 40-41, "...name is unique to the machine for that template.")

Regarding claim 20, Buxton teaches:

-instructions to set a value associated with the received identifier in the system storage equal to the number of lines of output stored in the system storage. (Rejection of claim 18 is incorporated and further claim contains limitations as recited in claim 12. Therefore claim 20 is rejected under the same rational as claim 12.)

Regarding claim 21, Buxton teaches:

Note: Buxton, col. 8, lines 51-53, disclosed command line processes as an alternative to a user interface.

-a processor; a storage device coupled to the processor; the storage device having stored thereon a program having instructions to receive an identifier, receive output from a command line utility, and store the command line utility output in the system storage at a location identified by the identifier. (See fig. 1. Claim 21 contains limitations as recited in claim 1, therefore claim 21 is rejected under the same rational as claim 1.)

Regarding claim 22, Buxton teaches:

-program comprises a dynamic link library. (Fig. 2, item 205.)

Conclusion

4. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman

01/19/2005

PETER WONG, DIRECTOR
TECHNOLOGY CENTER 2100